Natural Language Semantics via Logical Deduction

Mary Dalrymple
NUS, 3 April 2013

1 Language and linguistics

What is language?

- Language is an "instinct" (Pinker, 1994), reflecting innate mental structure and mental processes. Understanding language helps us understand the human brain. Explicit formal descriptions of language give a better and more detailed understanding of how the brain works.
- Language is a complex symbolic system in which basic structures (sounds, words) can combine according to a system of rules (a grammar) to form new structures (words, phrases). We can understand this complex system by identifying the basic components and the systems of rules that combine them.

What are the practical advantages of understanding the structure of language? Computational grammars allow practical applications involving language understanding and generation:

- Translation: a thorough understanding of the text is required, as well as knowledge of the grammar of the target language.
- Question answering: Parsing and understanding a body of text, and using this knowledge to answer questions from users. The California-based startup Powerset used a grammar of English and other languages to parse Wikipedia and build up a database of the knowledge contained there, and used the same grammars to parse user queries. Powerset was acquired by Microsoft for US$100 million in 2008.
- Natural language interfaces: Interact with your computer by speaking (or typing) to it, like in Star Trek. This requires rapid on-line processing of input, an understanding of how discourse works and what constitutes an appropriate response, and the ability to synthesize a grammatically acceptable response.

All of these practical applications depend on being able to compute the meaning of a sentence. This is a very complex problem. How can we determine the meaning of a sentence on the basis of the meanings of the words it contains?

2 Syntax and semantics: Traditional views

(1) Compositionality: The meaning of a sentence depends on the meanings of words and how those meanings are put together; the meaning of the whole depends on the meanings of the parts.

What are the relevant "parts"?

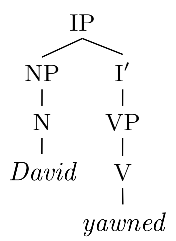
(2) Montague's view (e.g. Montague, 1973): for each syntactic (phrase structure) rule, there is a corresponding semantic rule.

Example phrase structure rule: a noun phrase (NP) can consist of a noun (N).

(3) NP $\longrightarrow$ N

Phrase structure tree for *David yawned*:

(4)



(5) Given meanings for “David” and “yawned”:

VP $\longrightarrow$ V: the meaning of the VP is the same as the meaning of the V
NP $\longrightarrow$ N: the meaning of the NP is the same as the meaning of the N
I′ $\longrightarrow$ VP: the meaning of the I′ is the same as the meaning of the VP
IP $\longrightarrow$ NP I′: the meaning of the IP is the result of *applying* the I′ meaning to the NP meaning

Advantage: This provides a systematic way of combining meanings which will work for every sequence of words that is a sentence according to our grammar of English; if our grammar of English produces a tree for a sentence, we can also produce a meaning for that sentence.

2.1 Expressing meanings: The lambda calculus

(6) We will use expressions of *predicate logic* to express meanings:

David, *yawn(David)*, *yawn*

(7) Function: when applied to an argument, yields a unique value.

yawn: applied to *David*, yields “true”.
applied to *Fred*, yields “true”.
applied to *George*, yields “false”.
...

(8) Function application:

yawn applied to *David* = *yawn(David)*

Sometimes functions correspond to individual words: for the sentence “David yawned”, we can get the meaning *yawn(David)* by applying *yawn* to *David* (and ignoring the past tense meaning of “yawned”). However, sometimes functions correspond to phrases consisting of more than one word: for the meaning “David selected Chris”, we would like to apply the meaning of “selected Chris” to the meaning for “David”. So we need a way of making a function which means “selected Chris”.

(9) Lambda abstraction:

$\lambda X.P$ represents a function from entities represented by X to entities represented by P.

Usually, the expression P contains at least one occurrence of the variable X, and we say that these occurrences are *bound* by the λ lambda operator.

(10) Function application:

$[\lambda X.P](a)$

The function $\lambda X.P$ is applied to the argument a.

Equivalent to the expression that results from replacing all occurrences of X in P with a.

Example: $[\lambda X.\mathit{yawn}(X)](\mathit{David}) \equiv \mathit{yawn}(\mathit{David})$

(11) Types

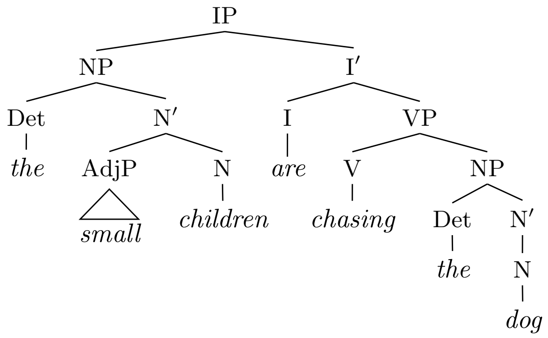
$\mathit{David} : e$	the constant *David* is of type e
$\mathit{yawn}(\mathit{David}) : t$	$\mathit{yawn}(\mathit{David})$ is of type t
$\lambda X.\mathit{yawn}(X) : \langle e \to t \rangle$	$\lambda X.\mathit{yawn}(X)$ is a function from things of type e to things of type t
$\lambda X.\lambda Y.\mathit{select}(X,Y) : \langle e \to \langle e \to t \rangle\rangle$	$\lambda X.\lambda Y.\mathit{select}(X,Y)$ is a function from things of type e to functions from things of type e to things of type t

3 Syntax: A more sophisticated view

Word order and phrase structure varies from language to language: it’s not a good guide for semantic composition.

3.1 English

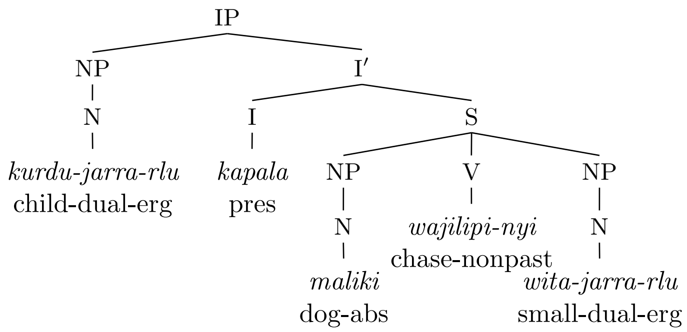
(12)



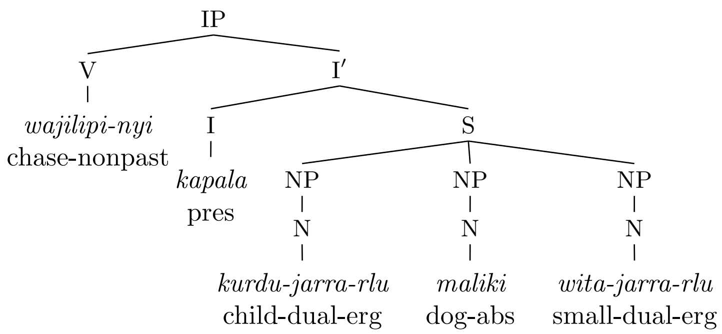
3.2 Warlpiri

As long as the auxiliary verb is in second position, any word order is acceptable.

(13) *kurdu-jarra-rlu kapala maliki wajilipi-nyi wita-jarra-rlu*
child-dual-erg pres dog.abs chase-nonpast small-dual-erg
'The two small children are chasing the dog.'



(14) *wajilipi-nyi kapala kurdu-jarra-rlu maliki wita-jarra-rlu*
chase-nonpast pres child-dual-erg dog.abs small-dual-erg
'The two small children are chasing the dog.'



3.3 Functional structures

In the linguistic theory of Lexical Functional Grammar (Bresnan, 2001; Falk, 2001; Dalrymple, 2001), there are two kinds of syntactic structure. The *phrase structure tree* (like 12–14) represents the order of words and how they group together. The *functional structure* or *f-structure* represents more abstract grammatical structure: grammatical relations like subject and object, modification relations, and information about tense and agreement.

(15) F-structure for English and Warlpiri ‘The small children are chasing the dog’:

$$\begin{bmatrix} \textsc{pred} & \text{‘}\textsc{chase}\langle\textsc{subj},\ \textsc{obj}\rangle\text{’} \\ \textsc{subj} & \begin{bmatrix} \textsc{pred} & \text{‘}\textsc{children}\text{’} \\ \textsc{spec} & \text{‘}\textsc{the}\text{’} \\ \textsc{mods} & \left\{ \begin{bmatrix} \textsc{pred} & \text{‘}\textsc{small}\text{’} \end{bmatrix} \right\} \end{bmatrix} \\ \textsc{obj} & \begin{bmatrix} \textsc{pred} & \text{‘}\textsc{dog}\text{’} \\ \textsc{spec} & \text{‘}\textsc{the}\text{’} \end{bmatrix} \end{bmatrix}$$

This uniformity makes the f-structure a promising level for stating rules of semantic composition; we can have similar composition rules for English, Warlpiri, and other languages if we state the rules at f-structure.

However, we can’t specify meaning instructions in terms of phrase structure rules, if the basis for semantic composition is f-structure.

Actually, we can do BETTER: we can use a formal logic to write instructions for putting together meanings on the basis of f-structure relations.

4 Glue: Composing meanings via deduction

Meaning assembly and linear logic (Dalrymple, 1999, 2001; Asudeh, 2004, 2005b; Kokkonidis, 2007):

- Logic of meanings (semantic level):
  the level of meanings of utterances and phrases
- Logic for composing meanings (‘glue’ level):
  the level responsible for assembling the meanings of parts to get the meaning of the whole

4.1 Choice of glue language

(16) Klein & Sag (1985, page 172):

> Translation rules in Montague semantics have the property that the translation of each component of a complex expression occurs exactly once in the translation of the whole. . . . That is to say, we do not want the set S [of semantic representations of a phrase] to contain *all* meaningful expressions of IL which can be built up from the elements of S, but only those which use each element exactly once.

(17) Linear logic: a resource logic *without* rules of weakening and contraction.

a. Weakening: we can include additional hypotheses in forming a conclusion.
If it is raining, you might get wet. It is raining. Therefore you might get wet.
If it is raining you might get wet. It is raining. It is Wednesday. Therefore, you might get wet.

b. Contraction: a hypothesis can be used any number of times.
If it is raining, you might get wet. If it is raining, you might catch a cold. It is raining. Therefore, you might get wet, and you might catch a cold.

(18) Linear logic and language:
Weakening unwanted: *Bill yawned Fred $\neq$ Bill yawned.
Contraction unwanted: *He thought that would win $\neq$ He thought that he would win.

(19) multiplicative conjunction: $\otimes$ (“and”)
linear implication: $\multimap$ (“implies”)

(20) INCORRECT: $A \vdash (A \otimes A)$

INCORRECT: $(A \otimes B) \vdash A$

CORRECT: $(A \otimes (A \multimap B)) \vdash B$

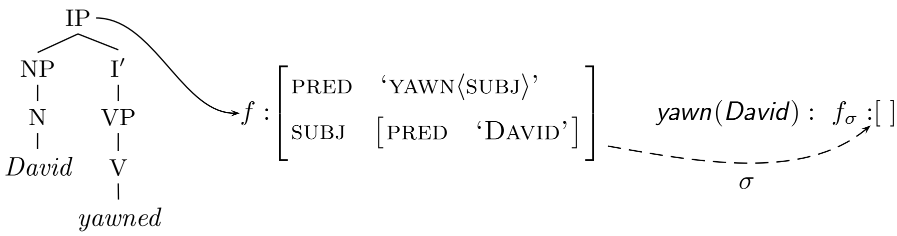
INCORRECT: $(A \otimes (A \multimap B)) \vdash (A \otimes B)$

INCORRECT: $(A \otimes (A \multimap B)) \vdash (A \multimap B) \otimes B$

4.2 Determining sentence meaning

We will assume that besides the two syntactic structures, the phrase structure tree and the f-structure, there is a semantic structure that is related to f-structure by the function σ.

(21) *David yawned.*

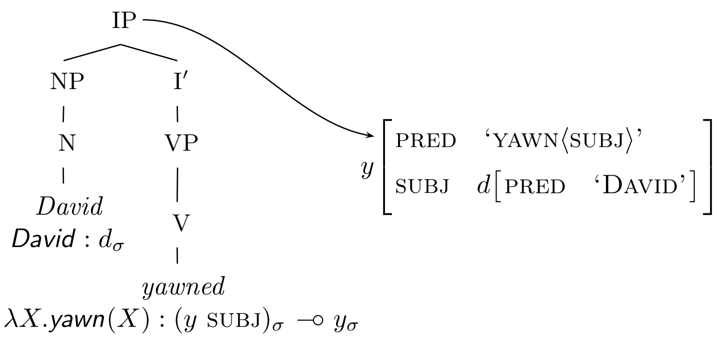


Meanings of words and how they combine with other meanings are specified by means of logical formulae; meaning composition then proceeds according to the rules of logic.

(22) $\mathit{yawn}(\mathit{David}) : f_\sigma$

5 Meaning specifications in the lexicon

(23) *David yawned.*



(24) Meaning constructors for *David yawned*:

David $\mathit{David} : d_\sigma$

yawn $\lambda X.\mathit{yawn}(X) : d_\sigma \multimap y_\sigma$

(25) Modus ponens in linear logic:

$$\frac{\begin{array}{l} f_\sigma \\ f_\sigma \multimap g_\sigma \end{array}}{g_\sigma}$$

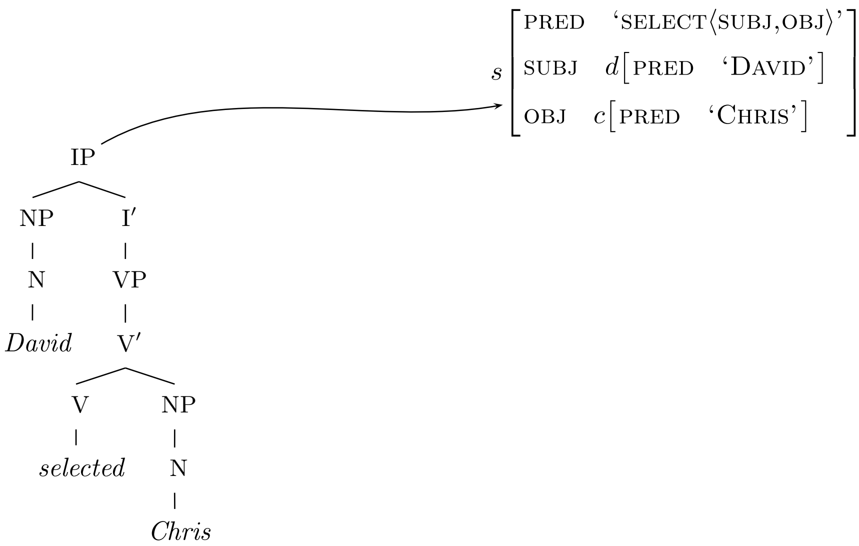
(26) Modus ponens in linear logic (the glue side) corresponds to function application in meaning logic (the meaning side):

$$\frac{\begin{array}{rcl} X & : & f_\sigma \\ P & : & f_\sigma \multimap g_\sigma \end{array}}{\begin{array}{rcl} P(X) & : & g_\sigma \end{array}}$$

(27)

$\mathit{David} : d_\sigma$	The meaning *David* is associated with the SUBJ semantic structure d_σ.
$\lambda X.\mathit{yawn}(X) : d_\sigma \multimap y_\sigma$	On the glue side, if we find a semantic resource for the SUBJ d_σ, we consume that resource and produce a semantic resource for the full sentence y_σ. On the meaning side, we apply the function $\lambda X.\mathit{yawn}(X)$ to the meaning associated with d_σ.
$\mathit{yawn}(\mathit{David}) : y_\sigma$	We have produced a semantic structure for the full sentence y_σ, associated with the meaning *yawn*(*David*).

(28) *David selected Chris.*



(29) Meaning constructor premises for *David selected Chris*:

David	David	$: d_\sigma$
Chris	Chris	$: c_\sigma$
select	$\lambda X.\lambda Y.\mathit{select}(X,Y)$	$: d_\sigma \multimap [c_\sigma \multimap s_\sigma]$

(30) $\mathit{David} : d_\sigma$ — The subject semantic structure d_σ is associated with the meaning *David*.

$\mathit{Chris} : c_\sigma$ — The object semantic structure c_σ is associated with the meaning *Chris*.

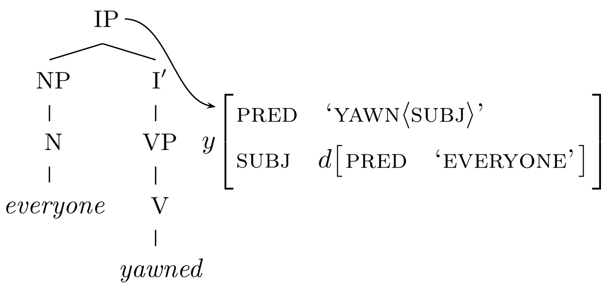
$\lambda X.\lambda Y.\mathit{select}(X, Y) : d_\sigma \multimap [c_\sigma \multimap s_\sigma]$

On the glue side, if semantic resources for the subject d_σ and the object c_σ are found, a resource for the sentence can be produced. On the meaning side, the two-place predicate *select* is applied to the subject meaning X and then the object meaning Y to produce the meaning $\mathit{select}(X, Y)$ for the sentence.

$\mathit{select}(\mathit{David}, \mathit{Chris}) : s_\sigma$ — We have produced a semantic structure s_σ for the full sentence, associated with the meaning $\mathit{select}(\mathit{David}, \mathit{Chris})$.

6 Quantification

(31) *Everyone yawned.*



(32) Meaning constructor premises for *Everyone yawned*:

everyone	$\lambda P.\ \forall X.\mathit{person}(X) \rightarrow P(X)$	$: \forall S.(e_\sigma \multimap S) \multimap S$
yawn	$\lambda X.\mathit{yawn}(X)$	$: e_\sigma \multimap y_\sigma$

(33) $\lambda X.\mathit{yawn}(X) : e_\sigma \multimap y_\sigma$ — On the glue side, we must find a semantic resource for the subject e_σ to produce a resource for the sentence, y_σ. On the meaning side, we apply the predicate to the subject meaning to produce the meaning for the sentence.

$\lambda P.\ \forall X.\mathit{person}(X) \rightarrow P(X) : \forall S.(e_\sigma \multimap S) \multimap S$

$\forall X.\mathit{person}(X) \rightarrow \mathit{yawn}(X) : y_\sigma$ — We have produced a semantic structure y_σ for the full sentence, associated with the meaning $\forall X.\mathit{person}(X) \rightarrow \mathit{yawn}(X)$.

7 Conclusion

The glue approach

- Exploits the ability of the logic to capture the f-structural syntactic constraints on semantic composition
- Enables a flexible interface between syntax and semantics: no unreasonable syntactic requirements imposed by theory of semantic composition; no unreasonable semantic requirements imposed by syntactic theory
- Exploits the resource-conscious properties of linear logic to ensure that each word contributes exactly once to the meaning of the sentence

Further reading:

- Dalrymple (1999): an early collection of papers. Warning: glue theory has not changed much since Dalrymple et al. (1993), but the notation has undergone a substantial change; the papers in Dalrymple (1999) use the old notation.
- Dalrymple (2001): an overview of LFG and glue, with analyses for a range of linguistic phenomena.
- Asudeh (2004, 2012): A glue-based analysis of copy raising and resumptive pronouns. Also see Asudeh & Crouch (2002a) on coordination, Asudeh & Crouch (2002b) on ellipsis, Asudeh (2005a) on control, Asudeh (2005b) on relational nouns, and other papers on Ash's web page: `http://users.ox.ac.uk/ cpgl0036/`
- Glue analyses have been incorporated into other linguistic theories besides LFG: for example, HPSG (Asudeh & Crouch, 2001) and TAG (Frank & van Genabith, 2001).

References

Asudeh, Ash. 2004. *Resumption as Resource Management*. Ph.D. thesis, Stanford University.

Asudeh, Ash. 2005a. Control and semantic resource sensitivity. *Journal of Linguistics* 41(3), pp. 465–511.

Asudeh, Ash. 2005b. Relational nouns, pronouns, and resumption. *Linguistics and Philosophy* 28(4), pp. 375–446.

Asudeh, Ash. 2012. *The Logic of Pronominal Resumption*. Oxford: Oxford University Press.

Asudeh, Ash & Richard Crouch. 2001. Glue semantics for HPSG. In Frank Van Eynde, Lars Hellan, & Dorothee Beermann (editors), *Proceedings of the 8th International Conference on Head-Driven Phrase Structure Grammar*, pp. 1–19. CSLI Publications. URL `http://cslipublications.stanford.edu/HPSG/2/hpsg01.html`.

Asudeh, Ash & Richard Crouch. 2002a. Coordination and parallelism in glue semantics: Integrating discourse cohesion and the Element Constraint. In Miriam Butt & Tracy Holloway King (editors), *On-Line Proceedings of the LFG2002 Conference*. CSLI Publications. URL `http://csli-publications.stanford.edu/LFG/7/lfg2.html`.

Asudeh, Ash & Richard Crouch. 2002b. Derivational parallelism and ellipsis parallelism. In *WCCFL 21: Proceedings of the 21st West Coast Conference on Formal Linguistics*. Medford, MA: Cascadilla Press.

Bresnan, Joan. 2001. *Lexical-Functional Syntax*. Oxford: Blackwell.

Dalrymple, Mary (editor). 1999. *Semantics and Syntax in Lexical Functional Grammar: The Resource Logic Approach*. Cambridge, MA: The MIT Press.

Dalrymple, Mary. 2001. *Lexical Functional Grammar*, volume 34 of *Syntax and Semantics*. New York: Academic Press.

Dalrymple, Mary, John Lamping, & Vijay A. Saraswat. 1993. LFG semantics via constraints. In *Proceedings of the 31st Annual Meeting of the ACL*, pp. 97–105. Columbus, OH: Association for Computational Linguistics.

Falk, Yehuda N. 2001. *Lexical-Functional Grammar: An Introduction to Parallel Constraint-Based Syntax*. Stanford: CSLI Publications.

Frank, Anette & Josef van Genabith. 2001. Glue Tag: Linear logic based semantics construction for LTAG — and what it teaches us about the relation between LFG and LTAG. In Miriam Butt & Tracy Holloway King (editors), *On-Line Proceedings of the LFG2001 Conference*. CSLI Publications. URL `http://csli-publications.stanford.edu/LFG/6/lfg1.html`.

Klein, Ewan & Ivan A. Sag. 1985. Type-driven translation. *Linguistics and Philosophy* 8, pp. 163–201.

Kokkonidis, Miltiadis. 2007. First-order glue. *Journal of Logic, Language and Information* 17, pp. 43–68.

Montague, Richard. 1973. The proper treatment of quantification in ordinary English. In Jaakko Hintikka, Julius Moravcsik, & Patrick Suppes (editors), *Approaches to Natural Language*, pp. 221–242. Dordrecht: D. Reidel. Reprinted in Thomason (1974, 247–270).

Pinker, Steven. 1994. *The Language Instinct: How The Mind Creates Language*. William Morrow and Company.

Thomason, Richmond (editor). 1974. *Formal Philosophy: Selected Papers of Richard Montague*. New Haven: Yale University Press.